

無線通信端末のパケットの深層学習を用いた解析

山本葵 (指導教員: 小口正人)

1 はじめに

近年、高性能になったスマートフォン、タブレット端末は世界中で増え続けている。気軽にネットワークにアクセスし、動画やゲームなどの大容量のデータ通信を楽しむことが出来るようになった。それに伴い、ネットワークにアクセスするトラフィックの量も年々増加している。大容量かつ高速な通信に対する需要は増大しているが、無線接続環境においては膨大なパケットが通信中に無線 LAN アクセスポイントに蓄積され、その結果輻輳が発生してしまうという問題も生じている。本研究では Android 端末を用いて無線 LAN 通信を行い、キャプチャしたアクセスポイント周りのパケットを用い深層学習を行なって解析する。無線 LAN 通信時のトラフィックの予測性能を評価した。

2 研究背景

世界中に増加し続ける高性能、高機能化したスマートフォンやタブレット端末は動画やゲームなどの大容量データ通信を容易にしている。それに伴い無線 LAN への負荷は増大すると考えられる。解決法としては、高速通信の規格化があり、可用帯域の増加や伝送速度の向上があるが、規格が広く普及するには時間がかかり、実際街中では狭い帯域を取り合っているのが現状である。

よって輻輳を早期検出しトラフィックを制御する必要がある。

3 先行研究

3.1 カーネルモニタ

先行研究で開発されたカーネルモニタ [1] は、通常見ることができない輻輳ウィンドウや RTT などの様々なカーネル内部 TCP パラメータをモニタできるツールである。このツールを Android に組み込むことで、TCP 通信時におけるカーネル内部の処理を、ユーザ空間からその様子をリアルタイムに解析し、各パラメータ値の変化を記録する。

3.2 輻輳制御ミドルウェア

カーネルモニタをベースとして、Android 端末間の連携した制御を目的とした輻輳制御ミドルウェア [2] がスマートフォン端末向けに開発された。これにより無線 LAN アクセスポイントにおける ACK パケットの蓄積を回避することができる。

また、[3] では複数の端末が同時に通信した時に、より全体の通信速度と公平性を向上することを目的として、この輻輳制御ミドルウェアに手を加え改良をおこなった。

4 深層学習

本研究では深層学習を行う。深層学習はニューラルネットワークの階層を深めたアルゴリズムで機械学習を実装するための 1 つの手法である。機械が自分自身で特徴量を抽出、階層を深めることで精度が大幅に向上した。代表的な実用例は、郵便局で郵便番号を認識して選別する際に使われる文字認識、Amazon の売り上げ

を大きくあげたことで有名な商品レコメンドシステムなどである。

4.1 LSTM(Long Short-Term Memory)

LSTM(図 1) は時系列データに対するモデルである。LSTM は隠れ層のユニットを LSTM block と呼ばれるメモリと 3 つのゲートをもつブロックにすることで実現された。その最も大きな特徴は文章などの長い依存関係にあるデータも覚えておくことができる長期依存が可能であるということである。

本研究では時系列データであるパケットの解析であるため、この LSTM をモデルとした深層学習を行う。

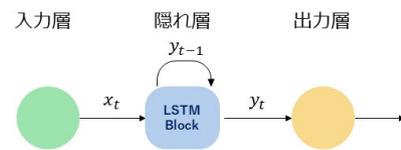


図 1: LSTM

5 実験

5.1 データセット

入力データとして無線 LAN アクセスポイント周辺のパケットを使用する。米国 riverbed 社のキャプチャデバイス、AirPcap を用いた [4]。AirPcap は wireshark 統合型のワイヤレストラフィックパケットキャプチャデバイスで、制御、管理、データの各フレームを含む、IEEE802.11 a/b/g/n のトラフィックをキャプチャできるデバイスである。

正解データとしては、輻輳を示す値を用意する。輻輳が発生していればアクセスポイントが混み合い、アクセスポイント宛の RTT 値は増加するのではないかとこの考えを元に、今回は PC とアクセスポイント間の RTT 値を使用した。

5.2 実験環境

実験環境を図 2 に示した。1 つのアクセスポイントに接続した複数台の Android 端末から iperf [6] によって、データをサーバに送信する。それと同時に PC で上記のデータを取得する。その後、適切なデータ形式に加工し、LSTM モデルを用いた深層学習をおこなう。

5.3 実験結果

以下の 2 つの実験を行なった。データを取得しその後、適切なデータ形式に加工し、深層学習用ライブラリとして Preferred Networks 社の Chainer [5] を用いて LSTM モデルで深層学習を行う。スマートフォン 2 台、タブレット 2 台の計 4 台の Android 端末を用いて 87 秒間通信を行いデータセットとした。2 つの実験は入力データが異なる。どちらの結果のグラフも t から t+9 秒の入力データを用いて t+10 秒の正解データである RTT 値を予測した。オレンジの線が正解の RTT 値、青の線が予測した値である。

5.3.1 結果 1(一次元入力データ)

第1段階の実験として表2の一次元の入力データで深層学習を行なった。全体的に振れ幅が少なく、平坦なグラフ図3が出力された。70秒付近の1番急激にRTT値が増加している部分は少し予測もついていっているが、これは十分に予測ができていたとはいえず、この程度の予測ではアクセスポイントが混み合っているかどうかの判断材料にすることはできないと考えられる。

5.3.2 結果 2(五次元入力データ)

次の実験は入力データを表3のように先ほどの一次元から五次元に増やし、精度の向上を目指して再度実験を行なった。図4のように結果は先ほどよりも良く、特に10秒付近のRTT値が急増している箇所や、65秒付近ではうまく予測が行われていることがわかる。

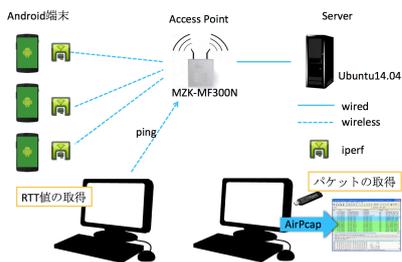


図2: 実験環境

表1: 実験機器の性能

Android	Model number	Nexus S	Nexus 7(2013)
	Firmware version	4.1.1	6.0.0
	CPU	1.0 GHz Cortex-A8	Quad-core 1.5 GHz Krait
	Memory(Internal)	16 GB, 512 MB RAM	16 GB, 2 GB RAM
	WLAN	Wi-Fi 802.11 b/g/n	Wi-Fi 802.11 a/b/g/n
server	OS	Ubuntu 14.04 (64bit) / Linux 3.13.0	
	CPU	Intel(R) Core(TM)2 Quad CPU Q8400	
	Main Memory	8.1GiB	
AP	Model	MZK-MF300N(Planex)	
	Support Format	IEEE 802.11 n/g/b	
	Channel	13	
	Frequency Band	2.4 GHz(2,1412-2,472 MHz)	

表2: 実験1 データセット

入力データ	正解データ
1秒間の平均データ量	RTT 値

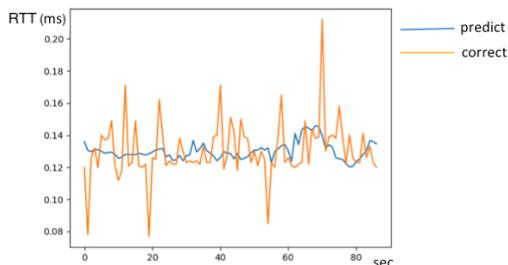


図3: 実験1 結果 (一次元入力データ)

6 まとめと今後の課題

本研究では、アクセスポイントに接続する端末が複数台通信を行い、輻輳がおこるであろう場合において、

表3: 実験2 データセット

1秒間	入力データ			正解データ
	時間	パケット数	平均データ量	RTT 値
		送信機器の台数	受信機器の台数	

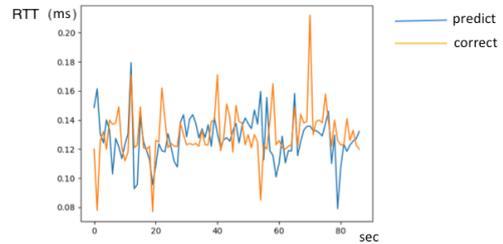


図4: 実験2 結果 (五次元入力データ)

輻輳の極めて早期な検出、予兆の発見を行うために、深層学習を用いてトラフィックの予測を行なった。入力データの情報を増やすことで精度が向上することが確認でき、トラフィックの予測には入力データの情報が非常に大きな意味を持つことがわかった。

今後の課題はさらに精度を向上させることである。今回は1秒間に流れてくる複数のパケットを集計して入力データとしたが、今後はその各パケット1つずつを入力データとして使用することを考える。また87秒間という短い時間のデータで実験を行なったが、数十分、数時間、数日とデータを集めて十分に学習をさせて実験を行う予定である。

謝辞

本研究を進めるにあたって、工学院大学の山口実靖先生より大変有用なアドバイスをいただきました。深く感謝いたします。

参考文献

- [1] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi: "Kernel Monitor of Transport Layer Developed for Android Working on Mobile Phone Terminals," Proc. ICN2011, pp.297-302, January 2011.
- [2] Ai Hayakawa, Saneyasu Yamaguchi, Masato Oguchi: "Reducing the TCP ACK Packet Backlog at the WLAN Access Point," Proc. ACM IM-COM2015, 5-4, January 2015.
- [3] Ayumi Shimada and Masato Oguchi: "A Study of Android Tables Performance," Proc.DEIM2017,H2-3,March 2017
- [4] riverbed, <https://www.riverbed.com>
- [5] Chainer,Framework for Neural Networks, <https://chainer.org/>
- [6] Iperf For Android Project in Distributed Systems, <http://www.cs.technion.ac.il/sakogan/DSL/2011/projects/iperf/index.html>